

**PX-0212**

**PX-0212**



**CONFIDENTIAL**

---

**Vendor Implementation  
Technical Specifications  
Punch Out Connectivity**

Date issued: September 9, 2003

---

Lawson Software Inc.  
380 St Peter Street  
St Paul, MN 55102  
[www.lawson.com](http://www.lawson.com)



L0002370  
CONFIDENTIAL

## CONTENTS

CONTENTS.....	2
Introduction .....	3
Application Flow.....	3
Vendor/B2B Partner Connectivity .....	3
Punch Out Connectivity .....	5
Punch Out Connectivity Diagram.....	5
cXML Header.....	6
XML Documents .....	6
Punch Out Setup Request.....	6
Punch Out Setup Response.....	7
Punch Out Order Message or PO Requisition.....	8
Purchase Order.....	11
Security .....	14

## Introduction

PunchOut for the purposes of this document is a term used to describe the ability of a Lawson end User to access a Vendor's website from within the Lawson procurement suite to generate a requisition. The requisition will consist of items selected on the Vendor's website, placed into a shopping cart and checked out. This document contains information that describes the internet Business to Business (B2B) trading specification between Lawson eProcurement and a Vendor with PunchOut capabilities. For the purposes of this document the relationship will be referenced between Lawson and the Vendor. Once the functionality between Lawson and the Vendor is proven then mutual customers of Lawson and the Vendor can utilize this in a production environment. The document provides potential Vendors with sample XML documents and a document flow diagram for the purposes of facilitating the establishment of PunchOut with Vendors.

## Application Flow

In a Punch Out scenario, the Vendor hosts the catalog and shopping cart. When a customer wishes to shop at the Vendor, they punch out from the Lawson eProcurement web page into the Vendors web site. When the request to shop at the Vendor's site is initiated by Lawson's eProcurement, Lawson builds a cXML PunchOutSetupRequest document and submits it to a predefined URL at the Vendor. The Vendor should parse the information and perform a background login of the user. The Vendor will then respond back to the Lawson eProcurement system with a cXML PunchOutSetupResponse document indicating whether or not the login was successful. If the login is successful, the response should contain a URL for Lawson eProcurement to use when linking directly to the Vendor. EProcurement will redirect the browser to the Vendor website using the URL supplied in the response. Once the link is made, the shopping experience will take place in the Vendor's website. The goal of the background login is to provide a seamless login for the customer, within Lawson, eProcurement to the Vendor.

After the link has been established and the browser is redirected to the Vendor's website, the customer can browse catalogs, check pricing, place items in a shopping cart, etc. If the customer decides to checkout the shopping cart contents, the Vendor will, upon checkout, build an XML PunchOutOrderMessage document (purchase order requisition) and post it to a designated URL on the Lawson eProcurement site. (The URL that was passed to the Vendor in the PunchOutSetupRequest document.). At this point an order has not been created but exists as a requisition within Lawson, awaiting approval. Once the requisition is approved, the order can be sent to the Vendor for processing as either an XML document or EDI transaction.

The Lawson technology allows websites to have the browser returned to a specific URL and the data to another. This allows the website performing the punchout to maintain the "state" of the web-user. It also treats the sending of the PunchOutOrderMessage as a background function that allows the data to be routed to a separate URL. This removes the need for using URL-Form-Encoding and allows for retry logic of the requisition data in the event the post were to fail

## Vendor/B2B Partner Connectivity

The following describes the steps involved to initiate integration with Lawson eProcurement and a Punchout Vendor highlighting the technical requirements.

### Integration Process:

- 1) Initial requirements definition – This is a high-level business requirement definition and begins to define the scope of the project.
- 2) Systems review – This step will be performed by team members from the Lawson B2B integration team along with team members from the B2B Partner. This process will familiarize each team with the system capabilities of the other. It also allows the Lawson team to further explain the Punchout process as it occurs within Lawson and to address any questions. The goal of this step is to flush out as many potential issues as possible prior to coding.
- 3) Clear up existing issues identified in the systems review.
- 4) Finalize the scope / requirements document and obtain sign off by all appropriate parties.
- 5) Once the scope is finalized, each team (Lawson and Vendor) will need to provide a detailed estimate of the work required.
- 6) Prioritization of the project by the Lawson B2B steering committee once the estimate is complete.
- 7) Steering committee sign off on estimate and Lawson's eProcurement Business Statement Of Work.

- 8) Vendor must assign USERID, PASSWORD and test URL for Lawson to test with.
- 9) Communicate implementation timeframe between Lawson and Vendor.

***Note: If the vendor currently supports the Lawson's existing dtd's and document flow then many of these steps are redundant and implementation would be immediate.***

Technical Requirements:

- 1) Lawson eProcurement operates across a secure website with firewalls, supporting SSL and 128 bit encryption.
- 2) Support Ariba's cXML PunchOut Request and Response.
- 3) Accept the PO Requisition (PunchOutOrderMessage).

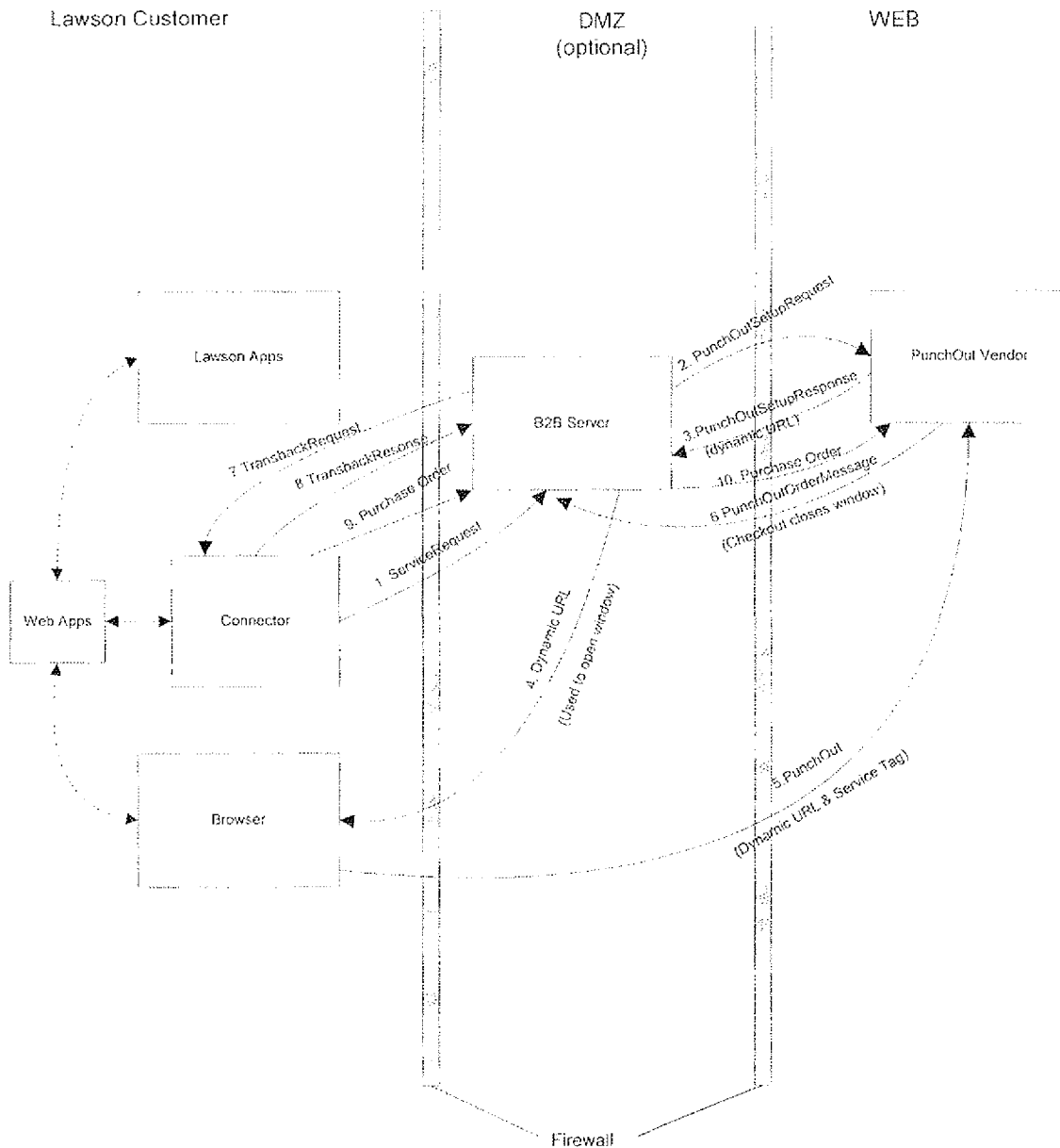
*Note: This list is not inclusive, but highlights some of the technical requirements*

## Punch Out Connectivity

The following diagram shows the interfaces between the Vendor and Lawson eProcurement for "PunchOut".

Note: Depending on the functionality of the website, the customer could choose to load the Requisition in their Lawson system and upon approval, bypass the Vendor's centralized customer website and use existing EDI interfaces to send the Purchase order. In this scenario, all EDI documents (confirmation, ship notice and invoice) would be returned directly to the Lawson Customer using existing EDI interfaces between the Lawson Customer and the Vendor.

## Punch Out Connectivity Diagram



## cXML Header

The Header element contains addressing and authentication information. The information in the Header element is the same regardless of which specific Request or Response is contained in the body of the XML message. The header element will be:

## XML Documents

The following are the XML Punch Out documents supported by Lawson:

- PunchOutSetupRequest (cXML)
- PunchOutSetupResponse (cXML)
- PunchOutOrderMessage (cXML)

The Lawson customer website will be able to send a PunchOutSetupRequest to the Vendor, and receive a PunchOutSetupResponse from the Vendor. The response will contain the specific URL and query string to be used by the Lawson customer website to redirect the browser to the Vendor website. The Vendor will validate all XML documents prior to processing. In order to accomplish this all XML documents sent from Lawson Customer and received by The Lawson Customer must have the SYSTEM attribute of the <DOCTYPE> tag formatted properly per the cXML user guide available at [www.cxml.org](http://www.cxml.org).

The following sections describe the cXML PunchOut transactions in further detail. This information is available at [www.cxml.org](http://www.cxml.org).

## Punch Out Setup Request

PunchOutSetupRequest is sent to the Vendor from the Lawson customer website.

Below is the dtd for the PunchOutSetupRequest.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v3.5 NT (http://www.xmlspy.com) by AN Other (Lawson Software) -->
<!-- DTD generated by XML Spy v3.5 NT (http://www.xmlspy.com)-->
<!ELEMENT BrowserFormPost (URL)>
<!ELEMENT BuyerCookie (#PCDATA)>
<!ELEMENT Credential (Identity, SharedSecret?)>
<!ATTLIST Credential
    domain CDATA #REQUIRED
>
<!ELEMENT From (Credential)>
<!ELEMENT Header (From, To, Sender)>
<!ELEMENT Identity (#PCDATA)>
<!ELEMENT PunchOutSetupRequest (BuyerCookie, BrowserFormPost)>
<!ATTLIST PunchOutSetupRequest
    operation CDATA #REQUIRED
>
<!ELEMENT Request (PunchOutSetupRequest)>
<!ELEMENT Sender (Credential, UserAgent)>
<!ELEMENT SharedSecret (#PCDATA)>
<!ELEMENT To (Credential)>
<!ELEMENT URL (#PCDATA)>
<!ELEMENT UserAgent (#PCDATA)>
<!ELEMENT cXML (Header, Request)>
<!ATTLIST cXML
    version CDATA #REQUIRED
    payloadID CDATA #REQUIRED
    timestamp CDATA #REQUIRED
>
```

Below is the cXML schema of the PunchOutSetupRequest:

```
<?xml version="1.0"?>
<!DOCTYPE cXML SYSTEM "http://xml.cxml.org/schemas/cXML/1.1.010/cXML.dtd"><cXML
version="1.0" payloadID="ac110160f70d9e3d0000476d2003-07-04T15:49:22-CDT172.17.1.96"
timestamp="2003-07-04T15:49:22-CDT" xml:lang="en-US">
  <Header>
    <From>
      <Credential domain="NetworkId">
        <Identity>YourUserID</Identity>
      </Credential>
    </From>
    <To>
      <Credential domain="DUNS">
        <Identity>Duns</Identity>
      </Credential>
    </To>
    <Sender>
      <Credential domain="AribaNetworkUserId">
        <Identity>YourIdentity</Identity>
        <SharedSecret>password</SharedSecret>
      </Credential>
      <UserAgent>Ariba ORMS 6.0</UserAgent>
    </Sender>
  </Header>
  <Request>
    <PunchOutSetupRequest operation="create">
      <BuyerCookie>ac110160f70d9e3d0000476d</BuyerCookie>
      <BrowserFormPost>
        <URL>http://172.17.1.96:5555/invoke/law.eps.process.receiveOrder</URL>
      </BrowserFormPost>
    </PunchOutSetupRequest>
  </Request>
</cXML>
```

## Punch Out Setup Response

The Vendor returns a PunchOutSetupResponse message. Below is the cXML schema for the response.

Below is the dtd for the PunchOutSetupResponse

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v3.5 NT (http://www.xmlspy.com) by AN Other (Lawson Software) -->
<!-- DTD generated by XML Spy v3.5 NT (http://www.xmlspy.com) -->
<!ELEMENT PunchOutSetupResponse (StartPage)>
<!ELEMENT Response (Status, PunchOutSetupResponse)>
<!ELEMENT StartPage (URL)>
<!ELEMENT Status EMPTY>
<!ATTLIST Status
  code CDATA #REQUIRED
  text CDATA #REQUIRED
>
<!ELEMENT URL (#PCDATA)>
<!ATTLIST URL
  name CDATA #REQUIRED
>
<!ELEMENT cXML (Response)>
<!ATTLIST cXML
  version CDATA #REQUIRED
  payloadID CDATA #REQUIRED
```



timestamp CDATA #REQUIRED  
reqCookie CDATA #REQUIRED

>

Below is the cXML schema for the PunchOutSetupResponse

```
<?xml version="1.0" encoding="UTF-8"?>
<cXML version="1.0" payloadID="2002-01-19T14:56:20.12349832.13284838@vendor.com"
timestamp="2002-01-19T14:56:20" reqCookie="cc4925e5f70d08c6000000b1">
  <Response>
    <Status code="200" text=""/>
    <PunchOutSetupResponse>
      <StartPage>
        <URL
name="VendorPage">http://punchout.com/rcomm/B2BStart.asp?sequence%5Fid=137194</URL>
      </StartPage>
    </PunchOutSetupResponse>
  </Response>
</cXML>
```

## Punch Out Order Message or PO Requisition

This interface is the requisition to be sent to the customer specified URL. Below is the cXML Schema for the PO Requisition.

Below is the dtd for the PunchOutOrderMessage.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v3.5 NT (http://www.xmlspy.com) by AN Other (Lawson Software) -->
<!-- DTD generated by XML Spy v3.5 NT (http://www.xmlspy.com)-->
<!ELEMENT Address (Name, PostalAddress, Email, Phone)>
<!ATTLIST Address
  isoCountryCode CDATA #REQUIRED
  addressID CDATA #REQUIRED
>
<!ELEMENT AreaOrCityCode (#PCDATA)>
<!ELEMENT BillTo (Address)>
<!ELEMENT City (#PCDATA)>
<!ELEMENT Country (#PCDATA)>
<!ATTLIST Country
  isoCountryCode CDATA #REQUIRED
>
<!ELEMENT CountryCode (#PCDATA)>
<!ATTLIST CountryCode
  isoCountryCode CDATA #REQUIRED
>
<!ELEMENT Credential (Identity, SharedSecret?)>
<!ATTLIST Credential
  domain (AribaNetworkUserId | DUNS) #REQUIRED
>
<!ELEMENT DeliverTo (#PCDATA)>
<!ELEMENT Description (#PCDATA)>
<!ATTLIST Description
  xml:lang CDATA #REQUIRED
>
<!ELEMENT Email (#PCDATA)>
<!ATTLIST Email
  name CDATA #REQUIRED
>
<!ELEMENT From (Credential)>
```

```

<!ELEMENT Header (From, To, Sender)>
<!ELEMENT Identity (#PCDATA)>
<!ELEMENT ItemDetail (UnitOfMeasure, Description, UnitPrice, URL, ManufacturerPartID,
ManufacturerName)>
<!ELEMENT ItemID (SupplierPartID, SupplierPartAuxiliaryID)>
<!ELEMENT ItemOut (ItemID, ItemDetail)>
<!ATTLIST ItemOut
    quantity CDATA #REQUIRED
    requestedDeliveryDate CDATA #REQUIRED
>
<!ELEMENT ManufacturerName (#PCDATA)>
<!ELEMENT ManufacturerPartID (#PCDATA)>
<!ELEMENT Money (#PCDATA)>
<!ATTLIST Money
    currency CDATA #REQUIRED
>
<!ELEMENT Name (#PCDATA)>
<!ATTLIST Name
    xml:lang CDATA #REQUIRED
>
<!ELEMENT Number (#PCDATA)>
<!ELEMENT OrderRequest (OrderRequestHeader, ItemOut)>
<!ELEMENT OrderRequestHeader (ShipTo, BillTo, Total)>
<!ATTLIST OrderRequestHeader
    orderDate CDATA #REQUIRED
    orderID CDATA #REQUIRED
    type CDATA #REQUIRED
>
<!ELEMENT Phone (TelephoneNumber)>
<!ATTLIST Phone
    name CDATA #REQUIRED
>
<!ELEMENT PostalAddress (Street, City, State, PostalCode, Country, DeliverTo)>
<!ATTLIST PostalAddress
    name CDATA #REQUIRED
>
<!ELEMENT PostalCode (#PCDATA)>
<!ELEMENT Request (OrderRequest)>
<!ATTLIST Request
    deploymentMode CDATA #REQUIRED
>
<!ELEMENT Sender (Credential, UserAgent)>
<!ELEMENT SharedSecret (#PCDATA)>
<!ELEMENT ShipTo (Address)>
<!ELEMENT State (#PCDATA)>
<!ELEMENT Street (#PCDATA)>
<!ELEMENT SupplierPartAuxiliaryID (#PCDATA)>
<!ELEMENT SupplierPartID (#PCDATA)>
<!ELEMENT TelephoneNumber (CountryCode, AreaOrCityCode, Number)>
<!ELEMENT To (Credential)>
<!ELEMENT Total (Money)>
<!ELEMENT URL (#PCDATA)>
<!ELEMENT UnitOfMeasure (#PCDATA)>
<!ELEMENT UnitPrice (Money)>
<!ELEMENT UserAgent (#PCDATA)>
<!ELEMENT cXML (Header, Request)>
<!ATTLIST cXML
    timestamp CDATA #REQUIRED
    payloadID CDATA #REQUIRED
>

```

Below is the cXML Schema for the PunchOutOrderMessage:

```
<?xml version="1.0" encoding="UTF-8" ?><!DOCTYPE cXML SYSTEM "cXML.dtd">
<cXML version="1.0" payloadID="2002-01-19T12:16:32.12349832.13284838@vendor.com"
timestamp="2002-01-19T12:16:32">
  <Header>
    <From>
      <Credential domain="DUNS">
        <Identity>62409873</Identity>
      </Credential>
    </From>
    <To>
      <Credential domain="AribaNetworkUserId">
        <Identity>YourUserID</Identity>
      </Credential>
    </To>
    <Sender>
      <Credential domain="DUNS">
        <Identity>62409873</Identity>
        <SharedSecret>nopassword</SharedSecret>
      </Credential>
      <UserAgent>Vendor cXML V1.1</UserAgent>
    </Sender>
  </Header>
  <Message>
    <PunchOutOrderMessage>
      <BuyerCookie>cc4925e5f70dff8d000000090</BuyerCookie>
      <PunchOutOrderMessageHeader operationAllowed="edit">
        <Total>
          <Money currency="USD">3386.00</Money>
        </Total>
      </PunchOutOrderMessageHeader>
      <ItemIn quantity="1">
        <ItemID>
          <SupplierPartID>220-8834</SupplierPartID>
          <SupplierPartAuxiliaryID>137186\1</SupplierPartAuxiliaryID>
        </ItemID>
        <ItemDetail>
          <UnitPrice>
            <Money currency="USD">2228.00</Money>
          </UnitPrice>
          <Description xml:lang="en">PentiumREG III Processor,1.13GHz with 15.0in UXGA
Display</Description>
          <UnitOfMeasure>EA</UnitOfMeasure>
          <Classification domain="UNSPSC">4317180100</Classification>
          <ManufacturerPartID>notused</ManufacturerPartID>
          <ManufacturerName>Vendor</ManufacturerName>
        </ItemDetail>
      </ItemIn>
      <ItemIn quantity="1">
        <ItemID>
          <SupplierPartID>220-5707</SupplierPartID>
          <SupplierPartAuxiliaryID>137186\2</SupplierPartAuxiliaryID>
        </ItemID>
        <ItemDetail>
          <UnitPrice>
            <Money currency="USD">1158.00</Money>
          </UnitPrice>

```

```

        <Description xml:lang="en">PentiumREG 4 Processor 1.8GHz, 256K Cache, Integrated NIC
and Sound</Description>
        <UnitOfMeasure>EA</UnitOfMeasure>
        <Classification domain="UNSPSC">4317180300</Classification>
        <ManufacturerPartID>notused</ManufacturerPartID>
        <ManufacturerName>Vendor</ManufacturerName>
    </ItemDetail>
</ItemIn>
</PunchOutOrderMessage>
</Message>
</cXML>

```

## Purchase Order

The Punchout Order Message or PO Requisition will be returned to the customer url (Lawson). An API will pass this back to the Lawson ERP and a Lawson Requisition will be created. This will pass through the normal approval process and a Purchase Order will be created. When the PO is issued, it will then pass through another API to create a cXML Purchase Order.

Below is the dtd for the Purchase Order

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v3.5 NT (http://www.xmlspy.com) by AN Other (Lawson Software) -->
<!-- DTD generated by XML Spy v3.5 NT (http://www.xmlspy.com)-->
<!ELEMENT Address (Name, PostalAddress, Email?, Phone?)>
<!-- ATTLIST Address
    isoCountryCode CDATA #REQUIRED
    addressID CDATA #IMPLIED
-->
<!-- ELEMENT AreaOrCityCode (#PCDATA)>
<!-- ELEMENT BillTo (Address)>
<!-- ELEMENT City (#PCDATA)>
<!-- ELEMENT Classification (#PCDATA)>
<!-- ATTLIST Classification
    domain CDATA #REQUIRED
-->
<!-- ELEMENT Country (#PCDATA)>
<!-- ATTLIST Country
    isoCountryCode CDATA #REQUIRED
-->
<!-- ELEMENT CountryCode (#PCDATA)>
<!-- ATTLIST CountryCode
    isoCountryCode CDATA #REQUIRED
-->
<!-- ELEMENT Credential (Identity, SharedSecret?)>
<!-- ATTLIST Credential
    domain (AribaNetworkUserId | DUNS | NetworkId) #REQUIRED
-->
<!-- ELEMENT DeliverTo (#PCDATA)>
<!-- ELEMENT Description (#PCDATA)>
<!-- ATTLIST Description
    xml:lang CDATA #REQUIRED
-->
<!-- ELEMENT Email (#PCDATA)>
<!-- ATTLIST Email
    name CDATA #REQUIRED
-->

```

```

>
<!ELEMENT Extrinsic (#PCDATA)>
<!ATTLIST Extrinsic
    name CDATA #REQUIRED
>
<!ELEMENT From (Credential)>
<!ELEMENT Header (From, To, Sender)>
<!ELEMENT Identity (#PCDATA)>
<!ELEMENT ItemDetail (UnitPrice, Description, UnitOfMeasure, Classification, ManufacturerPartID,
ManufacturerName, URL, Extrinsic+)>
<!ELEMENT ItemID (SupplierPartID, SupplierPartAuxiliaryID)>
<!ELEMENT ItemOut (ItemID, ItemDetail)>
<!ATTLIST ItemOut
    quantity CDATA #REQUIRED
    requestedDeliveryDate CDATA #REQUIRED
>
<!ELEMENT ManufacturerName (#PCDATA)>
<!ELEMENT ManufacturerPartID (#PCDATA)>
<!ELEMENT Money (#PCDATA)>
<!ATTLIST Money
    currency CDATA #REQUIRED
>
<!ELEMENT Name (#PCDATA)>
<!ATTLIST Name
    xml:lang CDATA #REQUIRED
>
<!ELEMENT Number (#PCDATA)>
<!ELEMENT OrderRequest (OrderRequestHeader, ItemOut+)>
<!ELEMENT OrderRequestHeader (Total, ShipTo, BillTo)>
<!ATTLIST OrderRequestHeader
    orderDate CDATA #REQUIRED
    orderID CDATA #REQUIRED
    type (new | update | delete) "new"
>
<!ELEMENT Phone (TelephoneNumber)>
<!ATTLIST Phone
    name CDATA #REQUIRED
>
<!ELEMENT PostalAddress (DeliverTo*, Street, City, State, PostalCode, Country)>
<!ATTLIST PostalAddress
    name CDATA #REQUIRED
>
<!ELEMENT PostalCode (#PCDATA)>
<!ELEMENT Request (OrderRequest)>
<!ATTLIST Request
    deploymentMode CDATA #REQUIRED
>
<!ELEMENT Sender (Credential, UserAgent)>
<!ELEMENT SharedSecret (#PCDATA)>
<!ELEMENT ShipTo (Address)>
<!ELEMENT State (#PCDATA)>
<!ELEMENT Street (#PCDATA)>
<!ELEMENT SupplierPartAuxiliaryID (#PCDATA)>
<!ELEMENT SupplierPartID (#PCDATA)>
<!ELEMENT TelephoneNumber (CountryCode, AreaOrCityCode, Number)>
<!ELEMENT To (Credential)>
<!ELEMENT Total (Money)>
<!ELEMENT URL (#PCDATA)>
<!ELEMENT UnitOfMeasure (#PCDATA)>
<!ELEMENT UnitPrice (Money)>

```

```

<:ELEMENT UserAgent (#PCDATA)>
<:ELEMENT cXML (Header, Request)>
<:ATTLIST cXML
  payloadID CDATA #REQUIRED
  timestamp CDATA #REQUIRED
>

```

Below is the cXML Schema for the Purchase Order

```

<?xml version="1.0" encoding="UTF-8" ?><!DOCTYPE cXML SYSTEM "cXML.dtd">
<cXML version="1.0" payloadID="2002-01-19T12:16:32.12349832.13284838@vendor.com"
timestamp="2002-01-19T12:16:32">
  <Header>
    <From>
      <Credential domain="AribaNetworkUserId">
        <Identity/>
      </Credential>
    </From>
    <To>
      <Credential domain="AribaNetworkUserId">
        <Identity/>
      </Credential>
    </To>
    <Sender>
      <Credential domain="AribaNetworkUserId">
        <Identity/>
      </Credential>
      <UserAgent/>
    </Sender>
  </Header>
  <Request deploymentMode="">
    <OrderRequest>
      <OrderRequestHeader orderDate="" orderID="">
        <Total>
          <Money currency=""/>
        </Total>
        <ShipTo>
          <Address isoCountryCode="">
            <Name xml:lang=""/>
            <PostalAddress name="">
              <Street/>
              <City/>
              <State/>
              <PostalCode/>
              <Country isoCountryCode=""/>
            </PostalAddress>
          </Address>
        </ShipTo>
        <BillTo>
          <Address isoCountryCode="">
            <Name xml:lang=""/>
            <PostalAddress name="">
              <Street/>
              <City/>
              <State/>
              <PostalCode/>
              <Country isoCountryCode=""/>
            </PostalAddress>
          </Address>
        </BillTo>
      </OrderRequestHeader>
    </OrderRequest>
  </Request>
</cXML>

```

```

<ItemOut quantity="" requestedDeliveryDate="">
  <ItemID>
    <SupplierPartID/>
    <SupplierPartAuxiliaryID/>
  </ItemID>
  <ItemDetail>
    <UnitPrice>
      <Money currency=""/>
    </UnitPrice>
    <Description xml:lang=""/>
    <UnitOfMeasure/>
    <Classification domain=""/>
    <ManufacturerPartID/>
    <ManufacturerName/>
    <URL/>
    <Extrinsic name=""/>
  </ItemDetail>
</ItemOut>
</OrderRequest>
</Request>
</cXML>

```

## Security

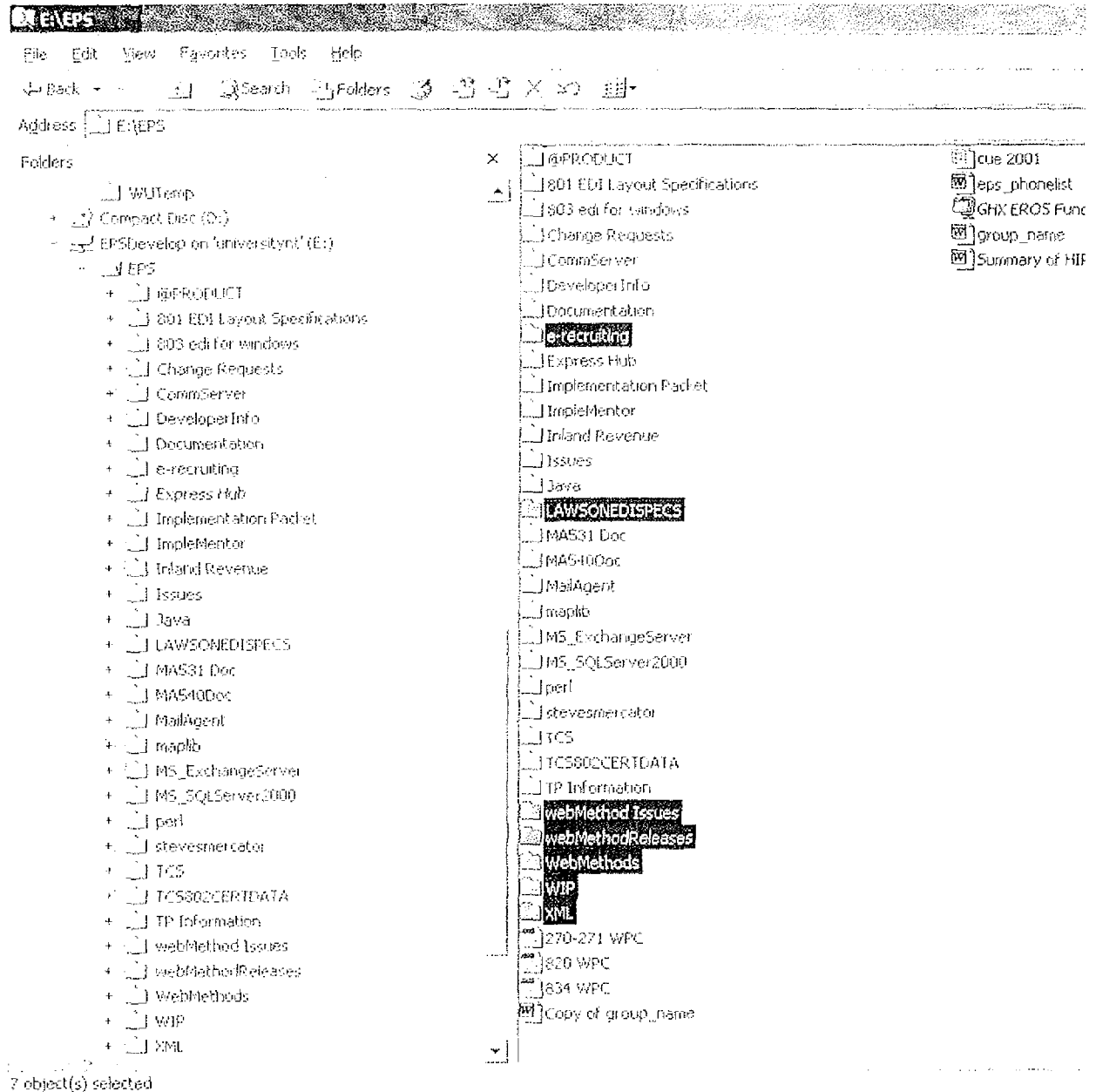
Below is a brief description of some of the security features.

- 1) Firewalls – Hardware and software that provides protection for unauthorized access into the internal network
- 2) Secured Socket Layers - Once the Lawson customer is logged into the website, the session is running under SSL.
- 3) Support for 128 bit encryption.
- 4) All Lawson customer databases reside behind the firewall.
- 5) Application level security. (Customers can see only their own data.)
- 6) Security logging capabilities at the server level.

Rob Wickham  
11/12/2003 02:50 PM

To: Matthew Braun/Lawson@Lawson  
cc: Doug Peterson/Lawson@Lawson  
Subject: Location of files

You will find most of the eProcurement and eRecruiting documents in the highlighted directories. Under WIP I have a directory robw that contains all my Requirements docs.



L0002384  
CONFIDENTIAL